

■ Smart■DARTS — Bewegung & Impact Detection

****Motion Gating und robuste Trefferdetektion (Land■and■Stick)****

■ Überblick

Die Pipeline besteht aus zwei Stufen:

- 1) ****MotionDetector**** (MOG2 + Morphologie) → aktiviert teure Schritte nur bei echter Bewegung.
- 2) ****DartImpactDetector**** (Konturen + Formmerkmale + zeitliche Bestätigung + Cooldown) → bestätigt einen Treffer stabil.

****Kerneffekte:**** CPU■Ersparnis durch Gating, robuste Erkennung auch bei Schatten/„Nachzittern“, keine Doppelzählungen.

■■ 1) Bewegungserkennung (MotionDetector)

****Algorithmus:**** `cv2.createBackgroundSubtractorMOG2` → Vordergrundmaske → Morph Open/Close → Konturenanalyse.

Ablauf

1. Hintergrundsubtraktion erzeugt `fg_mask` (bewegt vs. statisch).
2. Morphologie (Open/Close) reduziert Rauschen/Artefakte.
3. Größte Kontur → Schwerpunkts■/Flächen■/Intensitäts■Feature.
4. ****Gating:**** Nur wenn `*motion_pixel_threshold*` überschritten ist, wird die Impact■Detection getriggert.

Wichtige Parameter (`MotionConfig`)

| Parameter | Bedeutung | Praxiswerte |

| `var_threshold` | Empfindlichkeit MOG2 (je kleiner, desto empfindlicher) | 40–70 (50 default) |

| `motion_pixel_threshold` | Schwellwert in Pixeln für „Bewegung vorhanden“ | 300–1200 (500 default) |

| `min_contour_area` / `max_contour_area` | Konturgröße (px²), um z. B. Rauschen vs. Ganzkörperbewegung zu trennen | 100 / 5000 |

| `morph_kernel_size` | Strukturgröße für Open/Close | 3–5 |

| `event_history_size` | Anzahl gemerkter Motion■Events | 10 |

****Tuning■Hinweise:****

- Hell + geringer Bildrauschanteil → `var_threshold` höher; dunkler Raum → niedriger.

- Viele feine Artefakte → `morph_kernel_size` auf 5 heben.
- Zu viele unnötige Trigger → `motion_pixel_threshold` erhöhen.

■ 2) Impact Detection (DartImpactDetector)

Pipeline: Bewegungsmaske → Konturen → **FormFilter** → **CannyKanten** → **Score** → **MehrframeBestätigung** → **Cooldown**.

Form und Kantenmerkmale

- **Area** (px²): Größe der Kontur (gegen Staub/Noise begrenzen).
- **Aspect Ratio (AR)**: `w/h` – Darts sind länglich (typ. 0.3–3.0).
- **Solidity**: Fläche / konvexe Hülle (kompaktere Formen bevorzugt).
- **Extent**: Fläche / BoundingBox (Füllgrad).
- **Edge Density**: Kantenanteil in der Box (Canny 1/2).

Temporale Bestätigung: gleiche Position über `confirmation_frames` (z. B. 3) Frames → **Land and Stick**.

Cooldown: Nach Bestätigung ignoriert ein Kreis (`cooldown_radius_px`) die Umgebung für `cooldown_frames`, um Doppelzählungen zu verhindern.

Wichtige Parameter (`DartDetectorConfig`)

Gruppe	Parameter	Bedeutung	Praxiswerte
Größe	`min_area` / `max_area`	Konturfläche (px²)	10–1000
	Gestalt	`min_aspect_ratio` / `max_aspect_ratio`	Länglichkeit (w/h)
	`min_solidity` / `max_solidity`	Kompaktheit	0.10–0.95
	`min_extent` / `max_extent`	Füllgrad	0.05–0.75
Kanten	`edge_canny_threshold`	Canny-Schwellen	40 / 120
Gewichtung	`circularity_weight`, `solidity_weight`, `extent_weight`, `edge_weight`, `aspect_ratio_weight`	Scoring der Kandidaten	0.10–0.35 je nach Merkmal
Temporale Logik	`confirmation_frames`	stabile Frames bis Bestätigung	2–4 (3 default)
	`position_tolerance_px`	Positions-Toleranz	18–24
Cooldown	`cooldown_frames`	Frames ignorieren nach Treffer	25–40 (30)
	`cooldown_radius_px`	Ignorier-Radius (px)	45–55 (50)
Vorverarb.	`motion_mask_smoothing_kernel`	Gauß-Kernel (odd) für Otsu-Threshold	5–7

Presets (sinnvolle Startpunkte)

Preset	Charakter	Typischer Einsatz
--------	-----------	-------------------

| `aggressive` | sehr sensibel, findet mehr (mehr False Positives möglich) | Debug / langsames Video |

| `balanced` | Allround, robust | Standardbetrieb |

| `stable` | streng, höchste Präzision | Demo / Bühne / viel Publikum |

Beispiel:

```
python
```

```
from src.vision.dart_impact_detector import DartImpactDetector, DartDetectorConfig,
apply_detector_preset
```

```
cfg = apply_detector_preset(DartDetectorConfig(), "balanced")
```

```
det = DartImpactDetector(cfg)
```

```
---
```

■ End-to-End Ablauf (vereinfacht)

1. ROI-Frame → **MOG2** → `fg_mask`
2. **Morph Open/Close** → Rauschen weg
3. **Konturen** & Bounding-Boxes
4. **Form-Filter** + **Canny-Edge-Density** → Kandidaten Ranking
5. **Temporal Confirm** (\geq confirmation_frames)
6. **Cooldown** setzen → `cooldown_radius_px` für `cooldown_frames`
7. Impact speichern → **BoardMapper** → **Score**

```
---
```

■ Troubleshooting & Tuning

- **Fehlende Erkennung**: `min_area` senken, `var_threshold` senken, Licht erhöhen.
- **Zu viele False Positives**: `stable`-Preset, `confirmation_frames` erhöhen, `edge_` anheben.
- **Doppelzählungen**: `cooldown_radius_px` + `cooldown_frames` erhöhen.
- **Schwankende Position**: `position_tolerance_px` leicht erhöhen, ROI stabilisieren.
- **Zu hohe CPU**: ROI kleiner wählen, Gating-Schwellen erhöhen.

```
---
```

■ Quick-Reference (Defaults)

- Motion: `var_threshold=50`, `motion_pixel_threshold=500`, `morph_kernel_size=3`
- Impact: `confirmation_frames=3`, `cooldown_frames=30`, `cooldown_radius_px=50`, Canny `40/120`

■ Kurzfazit

Die Kombination aus **Motion■Gating** und **mehrstufiger Impact■Erkennung** liefert robuste Treffererkennung bei **hoher Echtzeit■Performance**. Mit Presets startklar, per Parametern fein justierbar für **Kamera**, **Licht** und **Distanz**.